
Métodos Computacionais para a Engenharia Eletrotécnica

Ano letivo 2021/2022

Miniprojeto

1. Introdução

Nesta secção são apresentadas as indicações gerais para a realização do miniprojeto no âmbito da unidade curricular de Métodos Computacionais para a Engenharia Eletrotécnica.

1.1 Avaliação

O miniprojeto tem uma cotação máxima de 4 (quatro) valores. Esta terá em conta o cumprimento dos objetivos propostos no presente enunciado, incluindo a avaliação do código e documentação, descritos mais à frente, e uma defesa oral.

1.2 Constituição de grupos

- Os alunos deverão formar grupos com **um máximo de dois elementos** para a realização das tarefas a desenvolver no âmbito do miniprojeto.
- É fortemente recomendado evitar a realização do presente trabalho de forma individual.
- Os grupos podem ser constituídos por alunos de turmas diferentes. Caso pertençam a turmas com docentes diferentes, deverão indicar qual o docente com o qual pretendem realizar a defesa oral do trabalho.

1.3 Submissão de miniprojeto

- O trabalho de cada grupo deverá ser submetido através do InforEstudante, através de uma submissão de trabalhos que será aberta atempadamente.
- O prazo de entrega do trabalho é até **às 23h59 do dia 28 de maio**.
- A entrega do miniprojeto consiste de:
 1. Todos os ficheiros *.m necessários para correr o miniprojeto desenvolvido;
 2. Um pequeno relatório (1-2 páginas) com a descrição do programa implementado, incluindo funções eventualmente desenvolvidas.

1.4 Defesa oral

- A defesa oral do miniprojeto é presencial, sendo **obrigatória a presença dos dois elementos de cada grupo**.
- A defesa do miniprojeto será agendada no InforEstudante, junto do docente que fará a respetiva avaliação, e terá a duração aproximada de 15 minutos. A inscrição dos alunos no InforEstudante é obrigatória para garantir uma melhor gestão da agenda.

- **A não comparência na defesa oral do miniprojeto corresponderá a uma classificação de zero valores para os alunos que faltarem.**
- A avaliação incidirá fundamentalmente sobre o cumprimento dos objetivos definidos para o miniprojeto, código desenvolvido e respetiva estruturação, verificações para evitar erros de utilização do programa, e clareza das justificações apresentadas às questões levantadas.

1.5 Situações de fraude

- Poderão ser consideradas situações fraudulentas os seguintes casos:
 - Plágio ou cópia autorizada de código desenvolvido por elementos externos ao grupo;
 - Cópia integral, parcial ou adaptada de código eventualmente disponível na Internet.
- Em relação ao último ponto, a utilização de código disponível na Internet é permitida desde que feito de forma limitada e sopesada. Nestes casos, e seguindo as boas práticas académicas, deverá ser feita a referência das fontes consultadas no relatório a entregar com o projeto (poderá considerar a inclusão de uma página adicional para as eventuais referências que venha a utilizar).
- **A deteção de uma situação de fraude será penalizada com a atribuição da classificação de zero valores ao miniprojeto.**

Métodos Computacionais para a Engenharia Eletrotécnica

Ano letivo 2021/2022

Miniprojeto

2. Descrição do miniprojeto

2.1 Objetivo

Pretende-se desenvolver um programa em MATLAB que permita fazer a animação da aplicação de um dos seguintes algoritmos de resolução de equações a uma variável independente: Método das bissecções, Método de Newton-Raphson e Método das secantes. Nas figuras 1 e 2 são ilustrados alguns exemplos de possíveis animações, sendo apresentados alguns instantes da animação da aplicação do método das bissecções e do método das secantes, respetivamente.

Na figura 1, onde é aplicado o método das bissecções, é perceptível o aparecimento de um ponto verde que marca o ponto obtido após a bissecção do intervalo que contém a solução da equação, intervalo que, caso o algoritmo esteja a convergir, será cada vez menor.

Na figura 2, é observa-se a aplicação do método das secantes, sendo possível nesta figura ver a marcação da primeira secante, a obtenção do ponto x_3 , e a marcação da secante seguinte.

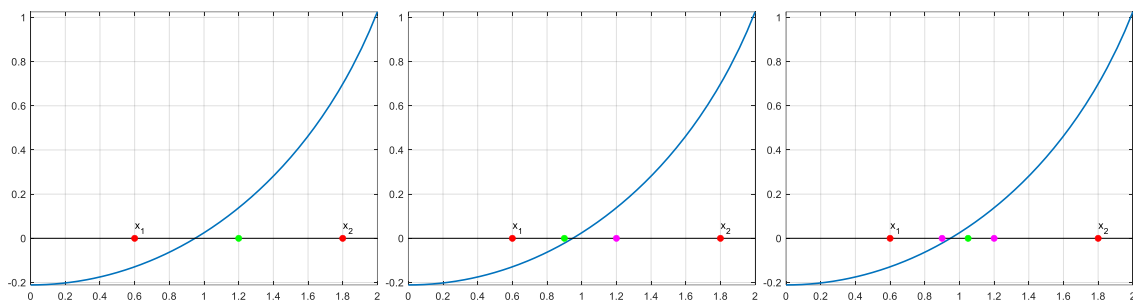


Figura 1. Exemplo demonstrativo de alguns instantes da animação a aplicação do método das bissecções.

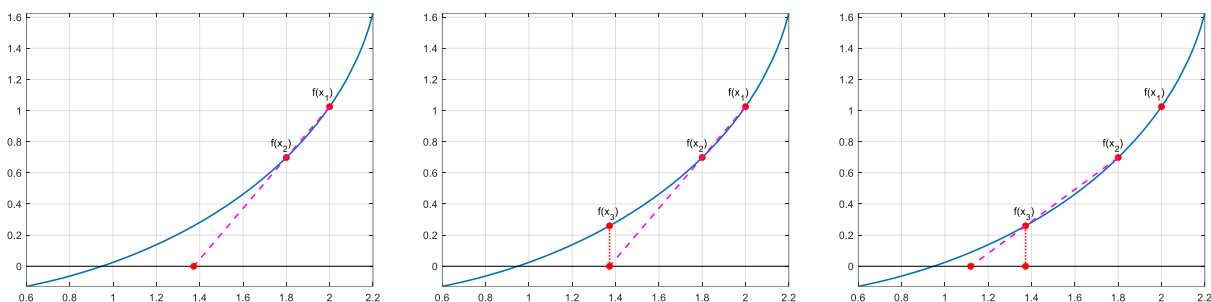


Figura 2. Exemplo demonstrativo de alguns instantes da animação a aplicação do método das secantes.

2.2 Descrição

De forma sumária, o funcionamento do programa a desenvolver no âmbito do miniprojeto é a seguinte:

- Deverá ter um menu inicial que permita ao utilizador seleccionar o método numérico a utiliza na resolução da equação. Este menu deverá incluir uma opção que permita ao utilizador sair do programa, i.e., o programa deverá correr até decisão do utilizador em terminar. Um possível exemplo de menu inicial é o seguinte:

```

*****
*****      Miniprojeto de MCEE 21/22      *****
*****      Resolução numérica de equações      *****
*****

```

```

[1] Mét das bisecções
[2] Mét. Newton-Raphson
[3] Mét. Secantes
[s] Sair

```

- Depois de selecionado o método, o utilizador deverá introduzir qual a equação a resolver, e os dados necessários para a resolução da mesma, tendo em conta o método selecionado.
- Definidos todos os dados necessários, o programa deverá gerar um gráfico animado, onde é demonstrada a aplicação do método selecionado.
- Ainda, deverá gerar uma tabela com informação dos dados em cada iteração (pelo menos, o número de iteração, e valor calculado da solução nessa mesma iteração).
- O programa deverá evitar que os dados introduzidos pelo utilizador originem erros de execução do programa (e.g., o programa não poderá parar devido a erro na introdução de uma opção diferente daquelas apresentadas no menu do exemplo acima apresentado).
- Adicional: o utilizador pode definir a tolerância da solução.
- Adicional: calcular o erro entre a solução numérica e a solução exata da solução, calculada, e.g., de forma simbólica.

2.3 Sobre a animação

Pretende-se gerar uma animação de forma simples, i.e., atualizar o gráfico a uma taxa de tempo constante, criando assim a ilusão de animação. No caso deste miniprojeto, elementos gráficos são apagados e adicionados de forma a demonstrar o funcionamento de cada um dos algoritmos propostos.

Em vez de criar um gráfico novo para cada novo *"frame"* da animação, o que seria oneroso em termos de tempo, pode considerar apenas apagar os elementos que entende necessários com a função **delete**. Para cada elemento que desenhe com o comando **plot**, pode obter o seu respetivo *handle*, i.e., um valor identifica unicamente aquele elemento, e, posteriormente, apenas apagar este elemento.

Considere que pretende criar a animação de um ponto que segue uma trajetória circular. A cada instante de tempo, poderá desenhar o ponto, apagando o que foi desenhado no instante anterior. A velocidade da animação pode ser controlada com uma pequena pausa (comando **pause**). Uma possível implementação é abaixo apresentada.

```

% Ponto a rodar no sentido anti-horário, sobre uma circunferência de raio
% unitário
%-----
% Coordenadas
%-----
t = linspace(0, 8*pi, 100); % oito voltas
x = cos(t);
y = sin(t);
t_pausa = 0.05; % Define o tempo de cada 'frame', neste caso 50 ms

%-----
% Gráfico
%-----

```

```

figure;
axis([-1.5 1.5 -1.5 1.5])
% A instrução seguinte garante que uma unidade em 'xx' seja igual a uma
% unidade em 'yy', e assim não distorcer a trajetória circular
set(gca, 'DataAspectRatio', [1 1 1])
grid on;
hold on; % necessário, senão perde-se a formatação dos eixos.

%-----
% Animação
%-----
% O plot devolve um 'handle', i.e., um identificador associado a unicamente
% à representação que é feita com esse comando plot
h = plot(x(1), y(1), 'ro', 'MarkerFaceColor', 'r', 'MarkerSize', 8);
shg
pause(t_pausa);

for i = 2:length(t)
    delete(h); % Apaga o ponto do instante anterior
    h = plot(x(i), y(i), 'ro', 'MarkerFaceColor', 'r', 'MarkerSize', 8);
    pause(t_pausa);
end

```

Apenas uma nota adicional sobre o comando **set**: esta instrução permite a alteração das propriedades dos gráficos, podendo alterar não apenas as curvas geradas, como os eixos (através de, por exemplo, o *handle gca* – *current axes*) ou mesmo da figura (através de, por exemplo, o *handle gcf* – *current figure*).

```

h = plot(0:pi/100:2*pi, sin(0:pi/100:2*pi));
set(h, 'color', 'r'); % muda a curva para vermelho
set(h, 'LineStyle', '--'); % linha a traço interrompido
set(h, 'LineWidth', 2); % espessura da linha a 2
set(gca, 'Xlim', [0 2*pi], 'Ylim', [-1.5 1.5]) % definição do eixos
set(gcf, 'position', [100 200 300 200]) % reposiciona e redimensiona a figura

```

Para mais informações pode consultar a documentação do MATLAB para os termos '*set*', '*gcf*', '*gca*', '*axes properties*', e '*figure properties*'.